

PJT명	프로젝트 기획과 협업	
단계	[Django 개발]	
진행일자	2025.11.07	
예상 구현 시간	필수기능	5H
	추가기능	2H
	심화기능	1H

1. 목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 Web application을 제작한다.
- Django Model과 ORM에 대하여 이해한다.
- Django Form에 대하여 이해한다.
- Many to one relationship(N:1)에 대하여 이해한다.
- 프로젝트를 기획하고, 기획 내용을 바탕으로 구체적인 요구사항을 작성할 수 있다.
- 기획서를 통해 개발 단계에서 참조할 수 있는 문서를 작성할 수 있다.

2. 준비사항

1) 제공 파일

- 팀별 기획서 양식 word 파일

2) 개발언어 및 툴

- Python 3.11
- HTML, CSS
- Visual Studio Code
- Google Chrome

3) 필수 라이브러리 / 오픈소스

- Django 5.2

- Bootstrap 5.3

3. 작업 순서

- 1) 팀원과 같이 요구사항을 확인하고, GitLab에 프로젝트를 생성한다.
 - 프로젝트 이름은 06-pjt로 지정한다.
 - 각 반 담당 강사님을 Maintainer로 설정한다.
- 2) 팀원과 합의하여 협업 방식을 결정하고, 요구사항을 구현한다.
- 3) 작성한 코드들을 정리하고, README를 작성한다.
 - .gitignore 파일을 활용하여 불필요한 파일 및 폴더는 제출하지 않는다.
- 4) README 작성이 완료되면 심화 학습을 진행한다.
- 5) 제출 기한에 맞춰 모든 산출물이 GitLab에 업로드 될 수 있도록 한다.

4. 요구사항

추천 알고리즘을 통한 영화 추천 커뮤니티 서비스를 구축하려고 한다. 다양한 스트리밍 플랫폼에서 제공되는 영화 정보를 수집 및 관리하고, 이를 기반으로 개인화된 영화 추천, 장르별 영화 탐색, 유사 영화 추천 등 다채로운 추천 기능을 설계 및 구현한다. 또한 영화에 대한 사용자 리뷰 및 감상평 공유 커뮤니티 기능을 제공하여, 사용자들이 활발하게 소통하고 정보를 교환할 수 있는 기능을 제공한다. 사용자는 자신이 본 영화를 평가하고, 다른 사용자의 리뷰를 참고하여 다음 영화를 선택하는데 도움을 받을 수 있다. 나아가, 관심 영화 목록을 맞춤형으로 구성하는 등 다양한 편의 기능을 제공한다. 팀원과 상의하여 아래 요구사항을 만족할 수 있도록 요구 사항 명세서를 작성 및 구현해보자.

영화와 관련된 여러 의견을 나눌 수 있는 기능을 만드는 단계이다. 영화 데이터의 생성, 조회, 수정, 삭제가 가능하며, 해당 영화에 대한 감상을 공유할 수 있는 댓글 기능이 구현된 애플리케이션을 완성해보자.

- 요구사항 예시(참고용)
 - 아래의 내용을 참고하여 추가적인 아이디어에 대해 요구사항을 추가 또는 수정하여 기능을 구현한다. 단, **필수 기능은 구현해야 하며, 수정할 수 없다.**

번호	분류	요구사항명	요구사항 상세	우선순위
기능적 요구사항				
F01	프로젝트	프로젝트 구성	영화 커뮤니티 서비스 구현을 위한 Django 프로젝트 및 앱 생성	필수
F02	movies	Model Class	영화 데이터를 데이터베이스에 저장할 수 있도록 Django Model 클래스 구현	필수

F03	movies	Model Class	영화에 대한 댓글 데이터를 데이터베이스에 저장할 수 있도록 Django Model 클래스 구현	필수
F04	movies	Form Class	사용자가 입력한 데이터 검증을 위한 Django Form 클래스 구현	필수
F05	movies view 함수	index	전체 영화 데이터를 조회하는 view 함수 구현	필수
F06	movies view 함수	create	영화 데이터를 입력할 수 있는 UI를 제공하고, 데이터에 따라 영화 데이터를 저장하는 view 함수 구현	필수
F07	movies view 함수	detail	단일 영화 데이터를 조회하는 view 함수 구현	필수
F08	movies view 함수	update	영화 데이터를 입력할 수 있는 UI를 제공하고, 선택한 영화 데이터를 수정하는 view 함수 구현	필수
F09	movies view 함수	delete	단일 영화 데이터를 삭제하는 view 함수 구현	필수
F10	movies view 함수	comments_create	댓글 데이터를 저장하는 view 함수 구현	필수
F11	movies view 함수	comments_delete	댓글 데이터를 삭제하는 view 함수 구현	필수
F12	AI 활용	포스터 저장	영화 데이터 생성 시 포스터 이미지를 업로드 할 수 있도록 구현	도전
F13	AI 활용	정보 생성 및 저장	생성형 AI API를 활용하여 감독 정보 검색 및 생성	도전

비기능적 요구사항

NF01	프로젝트 관리	Git 활용	개발자 간 작업 충돌이 일어나지 않게끔 Git을 활용하여 프로젝트 관리	필수
NF02	유지보수	URL 구성	URL이 변경될 때 유지보수가 용이하도록 구현	필수
NF03	보안	HTTP Method 허용	허용된 HTTP Method를 사용하는 요청만 허락하도록 구현	필수

1) 프로젝트 관리 (필수)

Git의 기능을 적극적으로 활용하여, 여러 개발자가 동시에 개발할 수 있는

환경을 조성한다.

- 요구사항 번호: NF01
- 팀원과 합의하여 branch 생성 원칙 정리하기
 - Gitflow, GitHub Flow 등을 참고하여 원칙을 정할 수 있음
 - branch 운영 중 문제상황이 발생할 경우 원인에 대한 분석을 꼭 진행할 것
- 팀원과 합의하여 commit 내역 기록 원칙 정리하기
 - 기능 개발, 버그 수정 등 그 목적이 명확히 드러날 수 있도록 commit 메시지를 작성할 것
 - 어느 시점에 commit을 남길지에 대하여 충분한 합의를 이룰 것
- 프로젝트 종료 시 초기에 세운 원칙이 잘 지켜졌는지 점검
 - README에 초기에 세운 원칙을 설명하여 정리할 것
 - 생성했던 branch 들에 대한 용도를 정리할 것
 - 작성된 commit들을 GitLab에서 확인하여 스크린샷으로 README에 포함시킬 것

2) 프로젝트 및 앱 (필수)

영화 데이터를 생성, 조회, 수정, 삭제할 수 있는 Django 프로젝트를 만든다. 명시된 요구사항 이외 서비스를 위해 필요하다고 생각되는 기능들은 자유롭게 구현해도 무관하며, Bootstrap을 활용하여 자유롭게 스타일링 한다.

- 요구사항 번호: F01, NF02, NF03
- 프로젝트 이름: mypjt
- 앱 이름: movies
- 페이지간 이동 편의성을 갖추기 위한 Navbar 구성
 - movies의 index, create의 기능들에 접근하기 편하도록 링크를 제공
 - 다양한 페이지에서 도달할 수 있도록 여러 페이지에 동일하게 출력되도록 구현

A. movies

영화 데이터를 관리하기 위한 앱이다.

- 요구사항 번호: F02, F03, F04
- Model 클래스
 - Movie: 영화 데이터를 나타내는 Model. 영화 제목, 줄거리, 감독, 장르, 평점을 저장할 필드 5개 지정
 - Comment: 영화에 대한 댓글을 나타내는 Model. 댓글 내용을 지정할 필드 1개 지정
 - Movie와 Comment는 1:N 관계를 맺고 있어야 함
- Form 클래스
 - 각 Model 클래스에 대하여 데이터 검증, 저장, 에러메시지 등을 관리하기 위해 적절한 ModelForm 사용
- 각 세부 기능들을 위한 view 함수를 구성
 - index, create, detail, update, delete, comments_create, comments_delete
 - URL이 변경될 가능성을 고려하여 유지보수에 유리한 형태로 구성
 - 각 view 함수가 허용하는 HTTP Method에만 호출되도록 구현

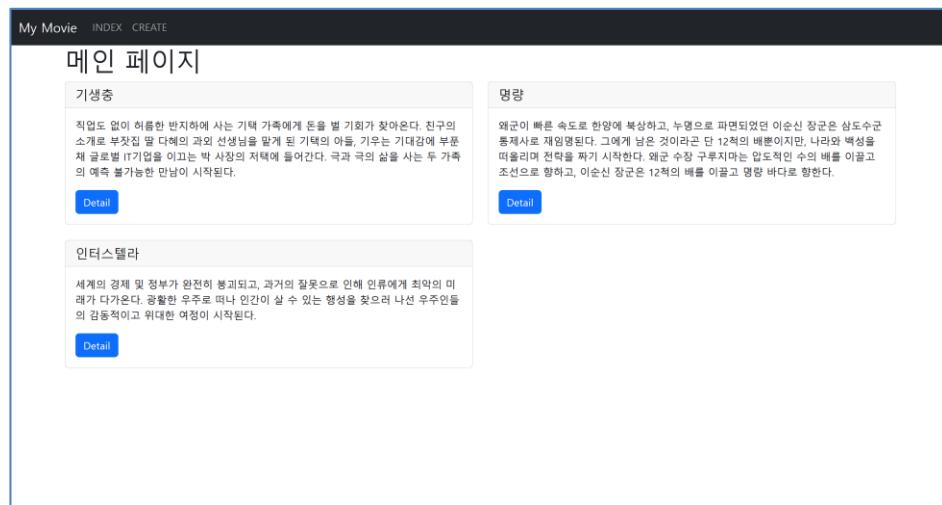
3) movies 앱 view 함수

영화 데이터를 생성, 조회, 수정, 삭제하는 역할을 하는 movies 앱을 구현하자.

A. index

전체 영화 데이터를 조회하기 위한 index view 함수를 구현한다.

- 요구사항 번호: F05
- 전체 영화 데이터가 조회되어야 함
- 특정 영화를 선택하여 detail 페이지로 이동할 수 있는 링크가 제공되어야 함
 - 링크의 문구와 UI는 자유롭게 구성
- 출력 예시



B. **create**

영화 데이터를 입력하기 위한 UI를 제공하고, 입력된 데이터를 저장하는 create view 함수를 구현한다.

- 요구사항 번호: F06
- Movie 모델의 데이터를 입력할 수 있는 UI 구현
 - 유효하지 않은 데이터가 저장되지 않도록 구현
 - 유효한 데이터의 경우 저장 후 작성된 영화의 detail 페이지로 redirect
 - UI 디자인은 자유롭게 구성
- 같은 view 함수에서 HTML 제공과 데이터 생성이 같이 이뤄지도록 구현
 - GET 요청을 받을 경우: HTML 제공
 - POST 요청을 받을 경우: 형식에 맞는 데이터는 저장, 형식에 맞지 않는 데이터는 오류를 페이지에 표시

● 출력 예시

- 장르는 3가지 중 택 1

My Movie INDEX CREATE

영화 등록 페이지

Title: 기생충

작업도 없이 허름한 반지하에 사는 기택 가족에게 돈을 벌기회가 찾아온다. 친구의 소개로 부잣집 딸 디해의 과외 선생님을 맡게 된 기택의 아들, 기우는 기대감에 부푼 채 글로벌 IT기업을 이끄는 박 사장의 저택에 들어간다. 극과 극의 삶을 사는 두 가족의 예측 불가능한 만남이 시작된다.

Description:

Director: 봉준호

Genre: Horror

Score: 5

제출



- 평점은 최소 0점에서 최대 5점이며 0.5점 간격으로 입력 가능

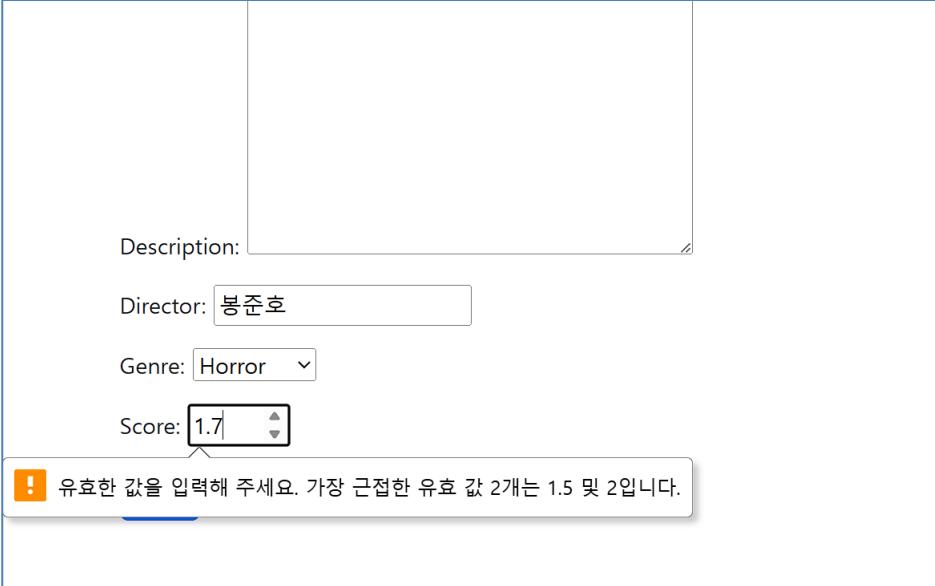
Description:

Director: 봉준호

Genre: Horror

Score: 1.7

! 유효한 값을 입력해 주세요. 가장 근접한 유효 값 2개는 1.5 및 2입니다.



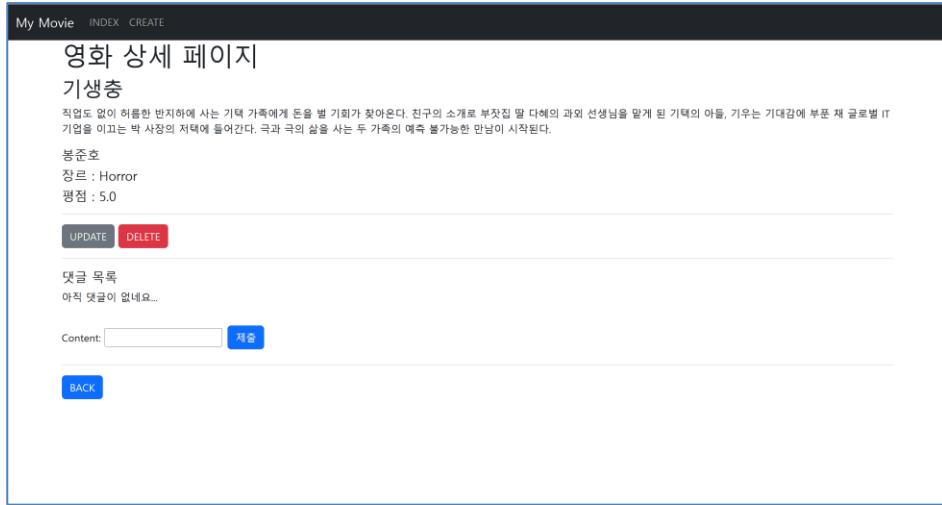
C. detail

단일 영화 데이터를 조회하는 detail view 함수를 구현한다.

- 요구사항 번호: F07
- index 페이지에서 영화를 선택했을 때 이동됨
 - 선택한 해당 영화의 정보를 바탕으로 내용이 출력되어야 함
- 영화 제목, 줄거리, 감독, 장르, 평점 데이터를 출력
 - 영화를 수정하기 위한 페이지로 이동하는 링크 구현
 - 영화를 삭제하는 버튼 구현
 - 이전 페이지(index)로 이동하는 링크 구현
 - UI 디자인은 자유롭게 구성
- 댓글 목록 및 작성 UI 출력
 - 작성된 댓글 목록이 보이도록 구현
 - 작성된 댓글을 삭제하는 버튼 구현
 - UI 디자인은 자유롭게 구성

● 출력 예시

- 댓글이 없을 경우



My Movie INDEX CREATE

영화 상세 페이지

기생충

직업도 없이 허름한 반지하에 사는 기택 가족에게 돈을 벌 기회가 찾아온다. 친구의 소개로 부잣집 말 다해의 과외 선생님을 맡게 된 기택의 아들, 기우는 기대감에 부푼 체 글로벌 IT 기업을 이끄는 박 사장의 저택에 들어간다. 극과 극의 삶을 사는 두 가족의 예측 불가능한 만남이 시작된다.

봉준호
장르 : Horror
평점 : 5.0

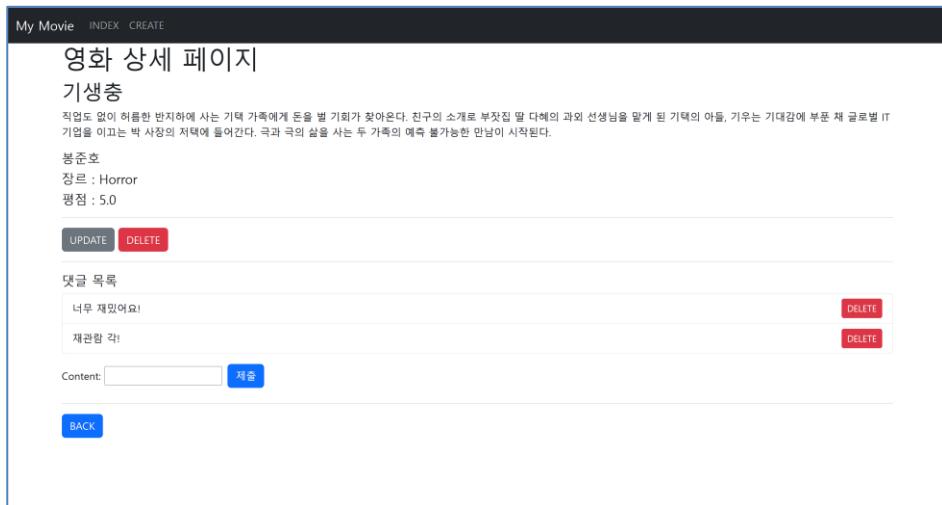
[UPDATE](#) [DELETE](#)

댓글 목록
아직 댓글이 없네요...

Content: [제출](#)

[BACK](#)

- 댓글이 있는 경우



My Movie INDEX CREATE

영화 상세 페이지

기생충

직업도 없이 허름한 반지하에 사는 기택 가족에게 돈을 벌 기회가 찾아온다. 친구의 소개로 부잣집 말 다해의 과외 선생님을 맡게 된 기택의 아들, 기우는 기대감에 부푼 체 글로벌 IT 기업을 이끄는 박 사장의 저택에 들어간다. 극과 극의 삶을 사는 두 가족의 예측 불가능한 만남이 시작된다.

봉준호
장르 : Horror
평점 : 5.0

[UPDATE](#) [DELETE](#)

댓글 목록

너무 재밌어요!	DELETE
재관람 각!	DELETE

Content: [제출](#)

[BACK](#)

D. update

영화 데이터를 수정하기 위한 UI를 제공하고, 입력된 데이터를 바탕으로 대상 영화 데이터를 수정하는 update view 함수를 구현한다.

- 요구사항 번호: F08
- detail 페이지에서 수정하기 링크를 통해 이동됨
 - detail 페이지에서 제공되었던 영화 정보를 바탕으로 기능이 동작되도록 구현
- Movie 모델의 데이터를 입력할 수 있는 UI 구현
 - 이전 영화의 데이터가 미리 입력되어 있도록 구현
 - 유효하지 않은 데이터가 저장되지 않도록 구현
 - 유효한 데이터의 경우 저장 후 수정된 영화의 detail 페이지로 redirect
 - UI 디자인은 자유롭게 구성
- 같은 view 함수에서 HTML 제공과 데이터 수정이 같이 이뤄지도록 구현
 - GET 요청을 받을 경우: HTML 제공
 - POST 요청을 받을 경우: 형식에 맞는 데이터는 저장, 형식에 맞지 않는 데이터는 오류를 페이지에 표시
- 출력 예시

The screenshot shows a web page titled "영화 수정 페이지" (Movie Update Page). At the top, there are navigation links: "My Movie", "INDEX", and "CREATE". Below the title, there is a "Title" input field containing the Korean text "기생충". A large text area follows, containing the following text:
직업도 없이 허름한 반지하에 사는 기택 가족에게 돈을 벌 기회가 찾아온다. 친구와 소개로 부잣집 딸 다예의 과외 선생님을 알게 된 기택의 아들, 기우는 기대감에 부푼 채 글로벌 IT기업을 이끄는 박 사장의 저택에 들어간다. 극과 극의 삶을 사는 두 가족의 예측 불가능한 만남이 시작된다.
Below this text area is a "Description" input field. Further down, there are fields for "Director" (Pongjunho), "Genre" (Horror selected from a dropdown menu), and "Score" (5.0). At the bottom right of the form is a blue "제출" (Submit) button.

E. **delete**

영화 데이터를 삭제하는 기능을 제공하는 delete view 함수를 구현한다.

- 요구사항 번호: F09
- detail 페이지에서 삭제하기 버튼을 통해 호출됨
 - detail 페이지에서 제공되었던 영화 정보를 바탕으로 기능이 동작되도록 구현
 - POST 요청에 대해서만 작동하도록 구현
 - 삭제가 된 이후에는 메인 페이지로 redirect

F. `comment_create`

댓글 데이터를 생성하는 기능을 제공하는 `comment_create` view 함수를 구현한다.

- 요구사항 번호: F10
- detail 페이지에 존재하는 댓글 작성 UI를 통해 호출됨
 - detail 페이지가 보여주고 있는 영화 데이터와 1:N 관계를 맺도록 구현
 - POST 요청에 대해서만 작동하도록 구현
 - 형식에 맞는 데이터는 저장, 형식에 맞지 않은 데이터는 오류를 페이지에 표시
 - 작성된 이후에는 관계를 맺고 있는 영화 데이터의 detail 페이지로 redirect
- UI는 detail 함수의 출력 예시를 참고하여 자유롭게 구현

G. **comment_delete**

댓글 데이터를 삭제하는 기능을 제공하는 comment_delete view 함수를 구현한다.

- 요구사항 번호: F11
- detail 페이지에 존재하는 댓글 삭제 UI를 통해 호출됨
 - POST 요청에 대해서만 작동하도록 구현
 - 작성된 이후에는 댓글이 관계를 맺고 있던 영화의 detail 페이지로 redirect

4) 도전 과제

생성형 AI 도구를 활용하여 도전과제 요구사항을 해결해보자.

- 코드 생성, 아이디어 구상, 문제 해결 방법 탐색 등 다양한 방식으로 활용할 수 있다.
- 사용할 생성형 AI 서비스는 자유롭게 선택한다.
 - 제공되는 GPT API Key를 활용할 경우, 모델은 반드시 gpt-5-nano 모델을 사용한다.
- 최종 결과물은 AI 생성 내용을 바탕으로 직접 수정 및 개선하여 적용해본다.

보조 수단으로 활용하되, 능동적인 자세로 학습에 임할 것.

- 최종적인 이해와 적용은 자기 주도적 학습을 통해 이루어지며, 배운 내용을 스스로 기록하고 정리하며 학습 효과를 높일 것.

A. 영화 포스터 이미지 업로드

게시글 작성 시 포스터 이미지를 업로드하여 각 게시글마다 포스터가 함께 출력되도록 구현해보자.

- 요구사항 번호: F12
- 게시글 작성 페이지에서 이미지를 추가할 수 있도록 구현
- 추가된 이미지는 메인 페이지와 영화 detail 페이지에서 확인할 수 있음
- 메인 페이지 레이아웃 UI는 자유롭게 구성
- 출력 예시
 - 생성 페이지



My Movie INDEX CREATE

영화 등록 페이지

Title: 기생충

Description:

작업도 없이 허름한 반지하에 사는 기택 가족에게 돈을 벌 기회가 찾아온다. 친구의 소개로 부잣집 딸 다혜의 과외 선생님을 맡게 된 기택의 아들, 기우는 기대감에 부푼 채 글로벌 IT기업을 이끄는 박 사장의 저택에 들어간다. 극과 극의 삶을 사는 두 가족의 예측 불가능한 만남이 시작된다.

Director: 봉준호

Genre: Horror

Score: 5

Poster img: 파일 선택 선택된 파일 없음

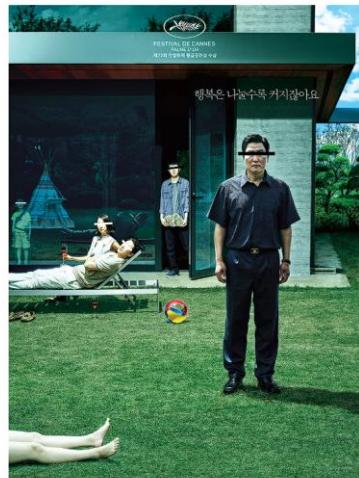
제출

- 상세 페이지



- 메인 페이지

메인 페이지



기생충

2019 훈종호 감독 작품 | 5월 30일 대개봉

기생충

직업도 없이 허름한 반지하에 사는 기택 가족에게 돈을 벌 기회가 찾아온다. 친구의 소개로 부잣집 밀 다혜의 과외 선생님을 맡게 된 기택의 아들, 기우는 기대감에 부푼 채 글로벌 IT기업을 이끄는 박 사장의 저택에 들어간다. 극과 극의 삶을 사는 두 가족의 예측 불가능한 만남이 시작된다.

[Detail](#)



명량

왜군이 빠른 속도로 한양에 복상하고, 누명으로 파면되었던 이순신 장군은 삼도수군통제사로 재임명된다. 그에게 남은 것이라곤 단 12척의 배뿐이지만, 나라와 백성을 떠올리며 전략을 짜기 시작한다. 왜군 수장 구루지마는 압도적인 수의 배를 이끌고 명량 바다로 향하고, 이순신 장군은 12척의 배를 이끌고 명량 바다로 향한다.

[Detail](#)

B. 생성형 AI API를 활용한 감독 정보 생성 및 저장

생성형 AI API를 활용하여 감독 정보를 생성하고 저장해보자.

- 요구사항 번호: F13
- 영화를 등록할 때 MovieForm으로 입력받은 감독 이름을 위키페이지 API의 쿼리로 전달해 감독 정보, 프로필 사진 검색
- 사용자 입력 데이터와 위키페이지 검색 정보를 바탕으로, GPT API를 활용하기
 - 감독 정보와 대표 작품 목록 요청
 - JSON 형태로 응답하도록 프롬프트 작성
- Movie 모델에 필드를 추가하여 저장
 - director_info와 director_works 필드에 추가
- 추가한 필드를 영화 detail 페이지에 출력
 - 감독 정보 영역 아래에 "감독 대표작과 감독 정보는 AI가 생성한 정보로, 정확하지 않을 수 있습니다."라는 안내 문구 추가
- 출력 예시

The screenshot shows a web-based movie management system. At the top, there's a navigation bar with 'My Movie', 'INDEX', and 'CREATE' buttons. Below the navigation, the title '영화 상세 페이지' (Movie Detail Page) is displayed, followed by '기생충' (Parasite). A descriptive paragraph about the movie follows. Below the text, there's a photo of director Bong Joon-ho and his name. A detailed bio is provided, mentioning his career from 1993 to 2020, including his debut film 'The Host' (2006) and his Oscar win for 'Parasite' (2019). At the bottom of the profile section, there are 'UPDATE' and 'DELETE' buttons. In the footer, there's a link to a '댓글 목록' (Comment List) and a message '너무 재밌어요!' (So funny!).

5. 프로젝트 기획

이번 단계에서는 관통 프로젝트(PJT)의 전체 설계 방향을 청사진 수준으로 기획하는 것을 목표로 한다.

지금까지 학습한 기술 스택(Django, DB, HTML/CSS, API, AI 활용 등)을 종합하여 팀별로 아이디어를 구체화하고, 실제 서비스로 발전 가능한 형태로 정리한다.

향후 진행할 관통 프로젝트의 데모를 확인하고, 이를 기반으로 서비스 아이디어를 도출한 뒤, **팀별 기획서(docx)를 작성하여 제출**해야 한다.

본 단계의 핵심은 단순한 코드 구현이 아니라,

- “어떤 문제를 해결하는 서비스를 만들 것인가?”
- “우리 팀의 서비스가 어떤 AI 기능과 결합될 수 있을까?”

를 명확히 정의하는 것이다.

제공되는 팀별 기획서 양식을 요약하면 다음과 같다.

단계	수행 내용	비고
1	관통 PJT 데모를 확인하고, 전체 서비스 흐름을 이해한다.	데모 사이트 참고
2	팀별로 어떤 서비스를 개발할지 기획한다. 기존 관통 PJT 주제를 응용하거나 확장하여, 우리 팀만의 차별화된 기능을 제안한다.	신규 기능 제안 가능
3	서비스에 적용할 AI 기능을 구상한다. 예를 들어, 추천 시스템, 자동 분석, 요약, 생성형 응답 등 다양한 방향을 검토한다.	OpenAI / Gemini / HuggingFace / 공공데이터 활용 가능
4	관련된 데이터셋을 탐색한다. (전자정부 데이터, 공공 API, 크롤링 등 활용 가능)	데이터 수집 방식 기획
5	유사 서비스나 레퍼런스를 조사하여	URL, 주요 기능 요약 포함

	벤치마킹한다.	
6	모든 내용을 '팀별 기획서.docx' 파일에 정리하고 GitLab에 업로드한다.	파일명: 팀번호_기획서.docx

6. 참고자료

- Django 5.2
<https://www.djangoproject.com/start/overview/>
- Python OpenAI Library
<https://platform.openai.com/docs/libraries?language=python>
<https://github.com/openai/openai-python>
- Excalidraw
손으로 그린듯한 스타일의 웹 기반 협업 화이트보드
<https://excalidraw.com/>
- diagrams.net (구 draw.io)
흐름도, ERD, UML 등 다양한 다이어그램을 온라인에서 무료로 작성할 수 있는 그래픽 편집 도구
<https://app.diagrams.net/?src=about>
- Figma
UI/UX 설계, 프로토타이핑, 팀 간 실시간 피드백을 지원하는 웹 기반 협업 디자인 툴
<https://www.figma.com/ko-kr/>
- Notion
노트, 문서, 위키, 프로젝트 관리 등을 하나의 워크스페이스에서 작성할 수 있도록 통합된 생산성 툴
<https://www.notion.com/ko>

7. 결과

제출 기한은 진행일 18시까지이므로 제출 기한을 지킬 수 있도록 한다. 제출은 GitLab을 통해서 이뤄진다.

● 산출물과 제출

- 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 상세히 기록한 README.md
- 완성된 각 문제 별 소스코드 및 실행 화면 캡쳐본
- 팀에서 설계한 설계문서 워드 파일 제출
- 프로젝트 이름은 06-pjt로 지정 및 각자 제출
 - 한 명의 GitLab 계정에 Git 저장소 Push 및 작업
 - 나머지 인원은 제출 시 해당 저장소를 Fork 하여 제출
- 각 반 담당 강사님을 Maintainer로 설정

- 끝 -